An area and power efficient on-the-fly LBCS transformation for implantable neuronal signal acquisition systems

Cosimo Aprile
LIONS and LSM, EPFL
Lausanne, Switzerland
cosimo.aprile@epfl.ch

Johannes Wüthrich
LSM, EPFL
Lausanne, Switzerland

Luca Baldassarre
LIONS and Gamaya
Lausanne, Switzerland

Yusuf Leblebici
LSM, EPFL
Lausanne, Switzerland

Volkan Cevher
LIONS, EPFL
Lausanne, Switzerland

ABSTRACT

A power and area efficient hardware encoding system tailored for wireless implantable applications is presented. Constant medical monitoring allowed by implantable devices is the most relevant alternative to current bulky monitoring systems, which, in case of severe mental diseases, require heavy surgery and long term hospitalization periods. In this work, the circuit design and the signal processing algorithm dovetail in order to allow real-time neuronal signal monitoring. Two main features must be met on the circuit level to facilitate the acceptance of the implant from the human body: small area and low power consumption. The presented work proposes a new compression scheme based on the Learning-Based Compressive Subsampling approach, which allows an area reduction with respect to recent published works, while allowing high signal reconstruction quality within low power requirements. The proposed method implements on-the-fly compression coefficients generation, which does not require large static memories. This new fully digital architecture handles the data compression of each individual neuronal acquisition channel with an area of $200 \times 190\mu m$ in 0.18 μm CMOS technology, and a power dissipation of only $1.15\mu W$.

Keywords

Compressive Sensing, neuronal signals, learning-based digital signal processing, area-efficient, low-power, signal recovery.

1. INTRODUCTION

Recent advances on micro-sized electrical technology is opening new possibilities to develop implantable systems for health-care applications. In particular, in case of some critical mental diseases treatment (such as drug resistant epilepsy), wireless implantable devices allow more flexibility for monitoring the electrical activity associated with a group of neurons. Implantable devices will allow to replace current medical surgery that requires days of hospitalization, passing through heavy and bulky medical monitoring equipment. Many efforts have been recently devoted to reduce the gap between the standard medical option and the future implantable solution. However, it is still necessary to address several challenges to make them more practical in terms of area and power dissipation. Considering multiple site recordings, the payload of data telemetry from the implant to the receiver station grows enormously.

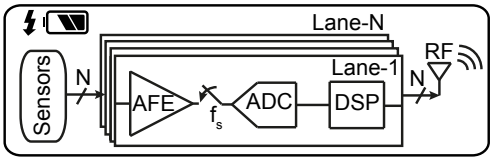

Figure 1: Typical block diagram of a multiple-channel implantable neuronal recording system.

Figure 1 shows a typical wireless implantable system with multiple channel sensors. In such device, the power delivered to the data telemetry block (the RF transmitter) is usually one order of magnitude higher than any other block of the implant, [1, 2]. In order to reduce such power bottleneck, the digital signal processing unit (DSP) is crucial to perform signal compression, drastically reducing the amount of data that is transmitted from the implant, while avoiding the loss of critical signal informations. In many recent approaches (e.g. [1,3–5] and references therein), a compression technique named compressive sensing (CS) has been implemented, to reduce the amount of data sampled by the implanted device. Indeed, CS allows to take less linear samples than standard sampling processes based on the Shannon-Nyquist theorem. Such simplified sampling process is possible since the information content of a signal is often much lower than its raw data content. However, in order to allow robust and high quality signal reconstruction, complex non-linear optimization problems have to be performed on the recovery node, which is then paid by latency and power requirements.

In this work, we present a fully digital encoder for neu-

CF '18, May 8–10, 2018, Ischia, Italy

DOI: https://doi.org/10.1145/3203217.3203260

ronal signals that applies a Learning-based Compressive Subsampling (LBCS) method [6], which is based on an on-the-fly compressed Hadamard transformation technique. Such method allows to drastically reduce the area requirements compared to previously published Hadamard-based LBCS [7], while still allowing the same signal reconstruction within a low power system implementation.

The paper is organized as follows: we introduce the main concepts of CS, LBCS and on-the-fly Hadamard generation are Section 2, while in Section 3, we describe the system implementation. The tailored circuit design for the dynamic Hadamard LBCS, is discussed in Section 4. Conclusions are drawn in Section 5.

2. COMPRESSION ALGORITHMS

In this section, we give a brief overview on the Compressive Sensing approach, highlighting its main advantages and disadvantages. Afterwards, we discuss the most recent Learning-Based Compressive Sampling, discussing its implementation within a tailored Hadamard transformation scheme.

2.1 Compressive Sensing

Given an input signal $\mathbf{x} \in \mathbb{R}^N$ which has K non-zero coefficients, Compressive Sensing (CS) states that $\mathbf{x}$ can be robustly recovered from a signal $\mathbf{y} \in \mathbb{R}^M$ containing fewer samples than dictated by the Shannon-Nyquist theorem, with $M = \mathcal{O}(K \log \frac{N}{K})$. The compressed version of the input signal $\mathbf{x}$ can be expressed as $\mathbf{y} = \mathbf{A}\mathbf{x} + \mathbf{w}$, where $\mathbf{A}$ is a linear operator that either satisfies the Restricted Isometry Property (RIP) or is incoherent [8], and $\mathbf{w}$ takes into account the measurement noise. If the input signal $\mathbf{x}$ is not sparse in the given domain, an ortho-normal basis $\mathbf{\Phi}$ has to be used to get a sparser representation of the original signal $\mathbf{x}$. Natural signals are often characterized by sparse and structured representations in time-frequency (or space-frequency) domains, such as provided by wavelets [9]. On the theoretical point of view, the $\mathbf{A}$ matrix can be generated with random coefficients, since i.i.d. sub-Gaussian matrices are incoherent and also satisfy the RIP condition. Moreover, they are universal, i.e., the RIP or the incoherence of $\mathbf{A}\mathbf{\Phi}$ is the same as of the original $\mathbf{A}$ [8], where matrix $\mathbf{\Phi}$ is used to move for a sparser representation of the signal $\mathbf{x}$. However, sub-Gaussian matrices are prohibitively expensive to use in practice, since they require $\mathcal{O}(MN)$ space and time. Being able to transmit only $\mathbf{y}$ allows to save on-chip storage and telemetry power. However, the reconstruction process needed to recover $\mathbf{x}$ from $\mathbf{y}$ requires to solve non-linear optimization problems that increase both time and power requirements on the recovery node.

Bernoulli (BERN) described in [1], Multi-Channel Sampling (MCS) [4] and Structured Hadamard Sampling (SHS) presented in [5] are randomized sampling approaches recently proposed for the compression of neural signals. These three architectures are very efficient on the sampling side, but require solving non-linear optimization problems to reconstruct the original signals.

As described in [10] and references therein, a reduced number of samples required for stable recovery can be achieved considering additional structures in the signal $\mathbf{x}$, such as interdependencies between its non-zero coefficients or constraints on its support, during the recovery process. As discussed in [5], the Hierarchical Group Lasso (HGL) approach gives the best performances over three different structured-sparsity recovery methods. Such approach has been used to compare the reconstructed iEEG signals sampled through BERN, MCS and SHS methods.

2.2 Learning-Based Compressive Subsampling

The compression method used in this work is based on the LBCS approach [6], which requires both linear encoding and decoding with respect to a given orthonormal basis. Such method allows to simplify both the sampling and signal restoring steps, compared to standard CS approaches. In a nutshell, LBCS can be summarized considering the following compression model $\mathbf{y} = \mathbf{P}_\Omega \mathbf{\Psi} \mathbf{x}$, where $\mathbf{\Psi} \in \mathbb{R}^{N \times N}$ is an orthonormal basis and $\mathbf{P}_\Omega \in \mathbb{R}^{M \times N}$ is a subsampling matrix, whose rows are canonical basis vectors. The effect of applying $\mathbf{P}_\Omega$ to $\mathbf{\Psi}\mathbf{x}$ is to retain only the coefficients indexed by the set Ω, also known as the *subsampling map*. The vector $\mathbf{y} \in \mathbb{R}^M$ is the compressed version of $\mathbf{x}$, with a nominal compression rate (CR) of $\frac{N}{M}$. The signal $\mathbf{x}$ is then approximately recovered via the fast linear decoder $\hat{\mathbf{x}} = \mathbf{\Psi}^* \mathbf{P}_\Omega^T \mathbf{y}$.

The learning process is dictated by a training set $\mathcal{D} = \{\mathbf{x}_1, \ldots, \mathbf{x}_m\}$ of m fully sampled signals of unit norm. The optimal subsampling map Ω is learnt by choosing the indices that capture most of the average energy in the transform domain:

$$\hat{\Omega} = \underset{\Omega, |\Omega| = M}{\arg\max} \frac{1}{m} \sum_{j=1}^{m} \sum_{i \in \Omega} |\langle \boldsymbol{\psi}_i, \mathbf{x}_j \rangle|^2, \tag{1}$$

where $\boldsymbol{\psi}_i$ is the i-th row of $\mathbf{\Psi}$. $\hat{\Omega}$ can be exactly found by selecting the M indices whose values of $\frac{1}{m} \sum_{j=1}^{m} |\langle \boldsymbol{\psi}_i, \mathbf{x}_j \rangle|^2$ are the largest [6]. The learnt sampling scheme is then used to directly sample only those transform coefficients indexed by $\hat{\Omega}$ for all signals $\mathbf{x}$.

Walsh-Hadamard based transformation has been used in recent publications [7, 11] because of its hardware friendly implementation, since each transformation coefficient requires one bit resolution, resulting in easy related computations. In particular, in [11] authors propose a threshold-based Walsh-Hadamard compression, to sample the Action Potentials (AP) related to neuronal signals for brain machine interfaces. The authors apply a butterfly scheme to transform the input signal samples into the Hadamard domain. However, such butterfly-based method can be performed on very few number of consecutive samples (8 samples in [11]), limiting any kind of learning approach because of the low signal statistic. For this reason, such work is used for AP signal detection, with limited implementation in constant medical monitoring for applications like epilepsy, where the whole signal behaviour is required by clinicians. Authors in [12] propose the generation of the full Hadamard matrix $\mathbf{\Psi} \in \mathbb{R}^{16 \times 16}$ for a parallel neural recording system. However, such implementation does not apply any compression mechanism, requiring an important power budget. The LBCS technique has been applied on circuit implementation with DCT-based transform [13]. Even though its implementation shows great signal reconstruction performances, the actual hardware implementation, which requires relatively larger area and power consumption with respect to its LBCS-Hadamard counterpart, makes it more suitable for different application, such as image processing. In [7], LBCS is exploited using the Hadamard transformation matrix. In such work, the whole Hadamard transformation matrix is stored in static memories which require more than 2/3 of the actual encoding area implementation.

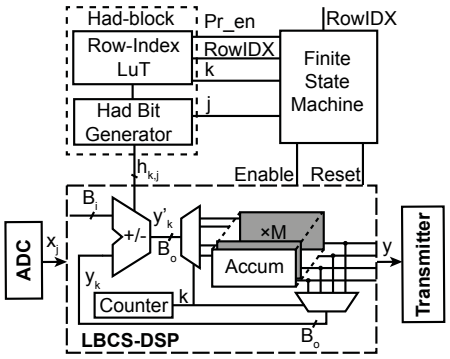

Figure 2: One channel block diagram showing the LBCS encoder and the matrix sequence generation logic.

In this work, we propose an LBCS based compression algorithm, which performs the transformation from temporal to Hadamard domain, through on-the-fly generated Hadamard coefficients. In this implementation, only the selected rows of the Hadamard matrix (defined by $\hat{\Omega}$) are generated and used for the embedded compression, resulting in a dynamic generation of the coefficients used to apply the LBCS approach. Such technique drastically reduces the encoder memory requirements needed by previous LBCS-Hadamard implementation, while the signal reconstruction quality is preserved within a low power chip implementation.

2.3 Walsh-Hadamard transformation

The Hadamard transform is particularly suited for hardware implementation since each coefficient can be computed by performing only simple additions or subtractions.

The reduction of hardware area in the Had-based LBCS described in [7] is possible by replacing the SRAM dedicated to store the Hadamard coefficients, with a direct computation of each matrix entry [12]. Such computation is feasible due to the intrinsic structure of the Hadamard matrix, which is summarized as follows. The non-normalized Hadamard transformation matrix $\hat{H}_n \in (-1,1)^{N\times N}$ of size n, with $N = 2^n$ is expressed as a recursive Kronecker product of two matrices

$$\hat{H}_n = \hat{H}_1 \otimes \hat{H}_{n-1}, \text{ where } \hat{H}_1 \triangleq \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} . \quad (2)$$

Each matrix coefficient indexes k and j, can be expressed in binary representation

$$k = \sum_{i=0}^{n-1} k_i 2^i \ , \ j = \sum_{i=0}^{n-1} j_i 2^i \ \text{ with, } k_i, j_i \in (0,1) . \quad (3)$$

Each Hadamard entry $h_{k,j}$ can then be expressed as

$$h_{k,j} = (-1)^{\sum_{i=0}^{n-1} k_i j_i} \equiv (-1)^{mod_2(\sum_{i=0}^{n-1} l_i j_i)} . \quad (4)$$

In particular, mapping the (1, -1) to (0, 1), each Hadamard entry can be derived by $h_{k,j} = mod_2(\sum_{i=0}^{n-1} l_i j_i)$. Such expression can be efficiently implemented in hardware, through logic AND gates to perform $l_i j_i$, while the module-2 sum is derived by a logic XOR. Thus, the circuit implementation takes the row and column indexes k and j and computes the Hadamard coefficient in the binary map (0, 1).

3. SYSTEM IMPLEMENTATION

The Hadamard-based LBCS encoder block diagram is depicted in Fig. 2, where is shown the input data path from the Analog to Digital Converter (ADC), through the LBCS Digital Signal Processor (DSP) to the encoded data transmitter. The *Finite State Machine* (FSM) of the DSP drives the *Had-block* and the main DSP core, where the encoding process is executed. The Had-block generates the Hadamard bit streams, and basically replace the SRAM used in previous implementation [7], reducing the encoder area requirement. The Had-block is mainly composed by the Row-Index Look up Table (LuT), and the Hadamard bit generator. The Row-Index LuT is meant to store the learnt indices of the sub-sampling matrix $\mathbf{P}_\Omega$, described in subsection 2.2. Assuming that only M rows of the full Hadamard matrix $\mathbf{H} \in \mathbb{R}^{N\times N}$ have to be used to apply the LBCS-based compression, then we can define a mapping function $w(k) = \in [0\ N-1]$, where $k \in [0\ M-1]$ is the index of the output value, and we define $h_{k,j} = h_{w(k),j}$. Then, the LuT implements such mapping function $w(k)$.

The LuT coefficients, driven by the FSM, are sent to the Hadamard-bit generator, which produces the transformation entries $h_{k,j}$, following the description done in subsection 2.3. During a calibration phase, the learnt Hadamard row indices, defined by the RowIDX input ($log(N)$ bit wide, to code all the possible Hadamard matrix indexes) are loaded in the LuT. As soon as the program enable (Pr_en) is active, the initialization starts and the FSM programs the M indexes into the LuT, following the RowIDX and the k signals used to address correctly the register. The FSM also generates and programs the enable and reset commands, sent to the DSP, to synchronize the encoding procedure correctly, and to reset at the end of each encoding window the accumulator registers (*Accum* in Fig. 2).

The encoder input signal x_j, digitized by the ADC with B_i bit resolution, is summed or subtracted from the previous accumulator register values, at each sampling instant j in the sampling window of length N. The LBCS-DSP block performs the embedded compression, defined as $y_k = \sum_{j=1}^{N} h_{k,j} x_j,\ k \in \{1,\ldots,M\}$, where $h_{k,j}$ is the (k,j)-entry of $\mathbf{H}_\Omega = \mathbf{P}_\Omega \mathbf{H}$; the Hadamard matrix $\mathbf{H}$ ($=\mathbf{\Psi}$ described in subsection 2.2), requires a single bit per entry, minimizing the computation costs in the transformation process. The encoder processing frequency is M times faster than the input signal frequency, in order to update each of the accumulator registers, where the transformation coefficients are stored.

3.1 System level trade-off

The previous Hadamard based LBCS implementation shown in [7], has been designed for sampling window of 256 samples (N = 256), with a fixed CR of 16×. In this work, we propose the hardware implementation with an on-the-fly Hadamard generation, with sampling window length N=64 and compression rate of CR=8. The same dataset as in [7] has been taken into account, to validate the proposed hardware implementation. The N=64 and CR=8 combination allows to get similar average reconstruction quality, while the LBCS encoder frequency f_s is halved, resulting in a lower power consumption. Indeed, since M is defined as N/CR, the larger is the number of the Hadamard rows M, the higher is the core LBCS clock frequency, which might become a limiting factor. On the other hand, a further reduction on the number of samples N, would degrade the signal statistics over which the learning approach is based on.

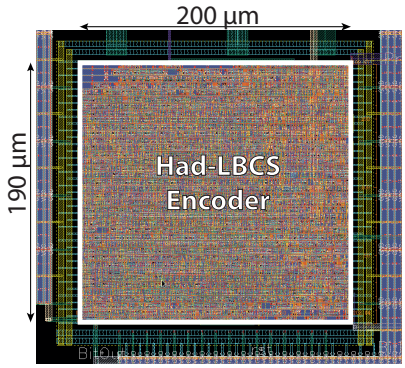

Figure 3: One-channel Had-LBCS encoder layout for $N = 64$ and $CR = 16$.

4. CIRCUIT DESIGN AND VALIDATION

The Had-LBCS system discussed in Section 3 and shown in Fig.2 has been designed in a 1P6M 0.18 μm CMOS technology. The layout of the fully digital implementation of the LBCS encoder is in Fig.3, highlighting the device sizes of 200×190 μm. It is worth noticing that such area consumption is smaller than the previous Had-LBCS work, even though the technology node is doubled (0.18 μm CMOS instead of 0.09 μm used in [7]).

A post layout simulation has validated the circuital implementation, verifying that the digitized neuronal signals given as input to the LBCS encoder are equal to simulations run off-line, on MATLAB. A worst case scenario with slow-slow process corner operating at 1.8 V has been used during the simulation process. Such analysis gives an estimated power consumption of the LBCS encoder of only 1.15 μW, which can be even further reduced in standard scenarios.

In order to have a one-to-one comparison with the previous Hadamard LBCS implementation [7], the equivalent of the presented work with N=256 and CR=16, has been designed in 0.09 μm CMOS technology. Such design verifies that the replacement of the SRAM with the on-the-fly Hadamard bit generator, reduces the layout area requirements of 20%.

5. CONCLUSIONS

This work presents an on-the-fly data compression system applying Hadamard-based LBCS approach. Such implementation allows to generate dynamically the matrix coefficients used for the compression algorithm, reducing drastically the area requirements of the encoding system, still maintaining the same reconstruction performances. Moreover, in a multichannel implementation, the Hadamard bit generation can be shared among all the neural channels, further reducing the overall implantable chip area requirements. For future implementations, the dynamic generation of the transformation coefficients might be used for variable CRs of each sampling window, depending on the signal characteristics.

Acknowledgment

This project has received funding from the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation programme (grant agreement 725594 - time-data). The authors would like to thank Jonathan Narinx (LSM, EPFL) for useful discussions on the system design.

6. REFERENCES

[1] F. Chen, A. P. Chandrakasan, and V. M. Stojanovic, "Design and analysis of a hardware-efficient compressed sensing architecture for data compression in wireless sensors," *IEEE Journal of Solid-State Circuits*, vol. 47, no. 3, pp. 744–756, 2012.

[2] S. Ha, A. Akinin, J. Park, C. Kim, H. Wang, C. Maier, P. P. Mercier, and G. Cauwenberghs, "Silicon-integrated high-density electrocortical interfaces," *Proceedings of the IEEE*, vol. 105, no. 1, pp. 11–33, 2017.

[3] J. N. Laska, S. Kirolos, M. F. Duarte, T. S. Ragheb, R. G. Baraniuk, and Y. Massoud, "Theory and implementation of an analog-to-information converter using random demodulation," in *IEEE International Symposium on Circuits and Systems*, 2007, pp. 1959–1962.

[4] M. Shoaran, M. H. Kamal, C. Pollo, P. Vandergheynst, and A. Schmid, "Compact low-power cortical recording architecture for compressive multichannel data acquisition," *IEEE Transactions on Biomedical Circuits and Systems*, vol. 8, no. 6, pp. 857–870, December 2014.

[5] L. Baldassarre, C. Aprile, M. Shoaran, Y. Leblebici, and V. Cevher, "Structured sampling and recovery of ieeg signals," in *6th IEEE CAMSAP conference*, 2015.

[6] L. Baldassarre, Y.-H. Li, J. Scarlett, B. Gözcü, I. Bogunovic, and V. Cevher, "Learning-based compressive subsampling," *IEEE Journal of Selected Topics in Signal Processing*, vol. 10, no. 4, pp. 809–822, 2016.

[7] C. Aprile, L. Baldassarre, V. Gupta, J. Yoo, M. Shoaran, Y. Leblebici, and V. Cevher, "Learning-based near-optimal area-power trade-offs in hardware design for neural signal acquisition," in *International Great Lakes Symposium On Vlsi (GLSVLSI)*. ACM, 2016, pp. 433–438.

[8] S. Foucart and H. Rauhut, *A mathematical introduction to compressive sensing*. Springer, 2013.

[9] S. Mallat, *A wavelet tour of signal processing*. Academic press, 1999.

[10] A. Kyrillidis, L. Baldassarre, M. El Halabi, Q. Tran-Dinh, and V. Cevher, "Structured sparsity: Discrete and convex approaches," in *Compressed Sensing and its Applications*. Springer, 2015, pp. 341–387.

[11] H. Hosseini-Nejad, A. Jannesari, and A. M. Sodagar, "Data compression in brain-machine/computer interfaces based on the walsh–hadamard transform," *IEEE transactions on biomedical circuits and systems*, vol. 8, no. 1, pp. 129–137, 2014.

[12] V. Majidzadeh, A. Schmid, and Y. Leblebici, "A 16-channel, 359 μw, parallel neural recording system using walsh-hadamard coding," in *Custom Integrated Circuits Conference (CICC), 2013 IEEE*. IEEE, 2013, pp. 1–4.

[13] C. Aprile, J. Wüthrich, L. Baldassarre, Y. Leblebici, and V. Cevher, "Dct learning-based hardware design for neural signal acquisition systems," in *ACM International Conference on Computing Frontiers 2017*, no. EPFL-CONF-228326, 2017.